

A TWO-DIMENSIONAL MESH REFINEMENT METHOD FOR PROBLEMS WITH ONE-DIMENSIONAL SINGULARITIES*

DAVID L. BROWN† AND LUIS GUILLERMO M. REYNA‡

Abstract. This paper introduces a method for resolving internal layers that can occur in the solutions of time-dependent differential equations in two space dimensions. Singular features in these solutions that are essentially one-dimensional in nature but are not oriented with the computational mesh are resolved using one-dimensional mesh refinement techniques with a procedure that is similar to an ADI method. A careful interpolation procedure assures that the resolution obtained in each ADI step is not lost in the succeeding ADI step.

Key words. two-dimensional mesh refinement, internal layers, shock resolution, alternating direction implicit method

1. Introduction. In this paper we discuss a numerical method developed to resolve internal layers which can occur in the solutions of time-dependent problems. As an example, the two-dimensional Burgers' equation exhibits behavior of the type we wish to discuss: We consider

$$(1.1) \quad U_t + \left(\frac{1}{2}U^2\right)_x + \left(\frac{1}{2}U^2\right)_y = \varepsilon(U_{xx} + U_{yy})$$

on $-\infty \leq x, y \leq \infty, t \geq 0$ with initial values $U(x, y, 0)$ given. Here $U = U(x, y, t)$ is the dependent variable, $0 < \varepsilon \ll 1$ is a small parameter and subscripts denote partial differentiation. Depending on the initial conditions given, the solutions of (1.1) can exhibit either boundary layer or internal layer behavior, i.e., regions of rapid change in the solution can occur in locally one-dimensional regions of width $O(\varepsilon)$, while the solution everywhere else varies on a length scale which is $O(1)$. The internal layers which occur in the solutions of (1.1) are typically referred to as *viscous shock profiles* with ε being the viscosity coefficient.

In some problems, such as compressible fluid flow, it is often the case that the accurate resolution of the viscous shock profiles is unimportant; only the size of the jump at the shock (and hence its speed) has any significant effect on the smooth part of the solution. Methods that give an accurate representation of the solutions to these types of problems without resolving the details of the shock structure can be found in, for example, Engquist and Osher [4], Osher and Solomon [12], Harten and Lax [6], and Chorin [3]. The purpose of this paper, however, is to present a method which can be applied to problems in which the detailed structure of the internal transition layers is important. Such problems arise, for example, in the study of chemically reacting fluids, where the details of the transitions can influence the speed of propagation of the reaction fronts. Methods for one-dimensional problems of this type have been considered by Brown [1], and Miller, Doss and Miller [11]. Brown uses adaptively determined local nonuniform moving mesh segments to resolve the moving features with rapid variation. This moving mesh is embedded within a much coarser mesh with meshwidths that are appropriate for accurately representing the smooth parts of the

* Received by the editors July 14, 1983, and in revised form January 23, 1984. This research was partially supported by the Office of Naval Research under contract N00014-80-CO076. Computer time was provided by the Stanford Exploration Project (Stanford University Department of Geophysics), the National Center for Atmospheric Research, and on the California Institute of Technology Applied Mathematics Department Fluid Dynamics VAX.

† Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125.

‡ Courant Institute of Mathematical Sciences, New York University, New York, New York 10012.

solution. It is essentially this method that will be extended in this paper to the two space-dimensional case. We will restrict our consideration to problems whose solutions exhibit rapid transitions that are essentially one-dimensional in nature, such as viscous shock profiles. For brevity, we will often refer to such a feature as a "shock", but with the understanding that for the reasons discussed above, we are always interested in its viscous profile.

The method for two-dimensional mesh refinement that we will discuss in the following sections was originally suggested to us by H. O. Kreiss [10]. It uses the one-dimensional finite difference approximation and mesh refinement procedure discussed by Brown [1] within a "splitting" or ADI procedure to solve problems in two space dimensions. The feature of this algorithm that makes it unique among mesh refinement procedures is that one-dimensional mesh refinement techniques are applied directly to two-dimensional problems. A careful interpolation procedure is used to transfer information from the grid lines in one direction to those in the other direction in the calculation without degrading the resolution of the solution obtained on each set of one-dimensional meshes. This method is explained and discussed in more detail in §§ 2 through 4. Numerical examples illustrating the method are included in § 5.

2. The mesh refinement procedure. For a problem in two space dimensions in which the shock line is very nearly linear and oriented so as to be parallel to one set of coordinate lines, it is clear how to implement a mesh refinement in order to resolve its viscous profile. If, for example, the shock lies essentially parallel to a line $x = x_0$, a refinement in the direction normal to the shock (the x -direction) could be made (see Fig. 1). No refinement would probably be necessary in the y -direction in this case. It

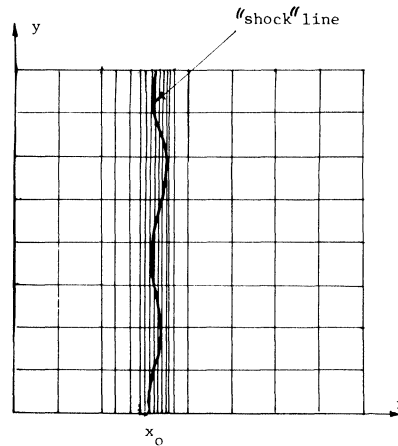


FIG. 1

is clear, however, that this will not always be true. We will not always have the freedom to choose the orientation of the computational mesh in such a way as to have "one-dimensional" rapid transitions oriented with the mesh. If the singular domain associated with the shock was not oriented with the mesh, then adding lines to refine the mesh would result in lines being added in both the x and y directions. In particular, we would also be refining the mesh in regions where the solution is smooth (see Fig. 2). This large number of added points in the mesh can clearly be reduced if we truncate the added lines so that they do not extend into smooth parts of the solution (see Fig. 3). The reduction in the number of meshpoints will, however, be at the expense of programming complexity.

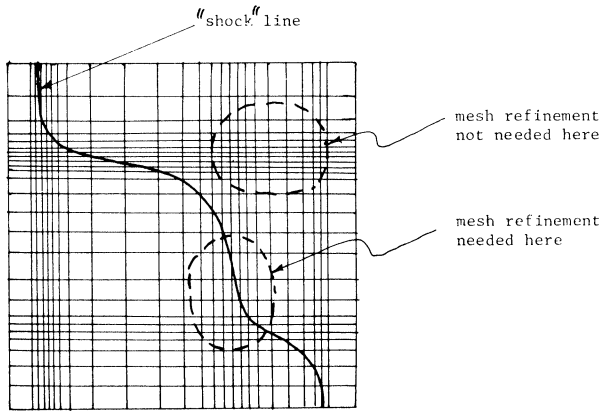
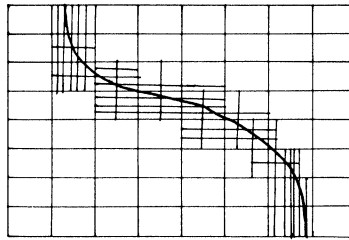
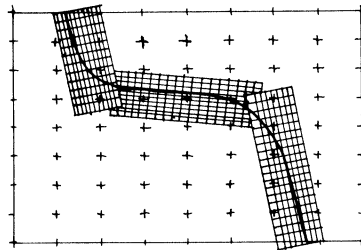


FIG. 2

FIG. 3. *Truncated mesh lines.*

A more rational approach to local mesh refinement is that of *composite meshes*. In this technique, local oriented meshes are embedded in the coarse mesh and interpolation is used to couple the solutions on the different meshes when solving the differential equations. Berger and Oliger [2], and Gropp [5] have used local oriented rectangular moving grids to accomplish this. Figure 4 illustrates the basic idea. With a simple

FIG. 4. *Method of Berger, Oliger and Gropp.*

extension of the grid generation approach of B. Kreiss [8], curvilinear grids could also be embedded in the coarse rectangular grid in such a way as to resolve a shock (see Fig. 5). Again interpolation would be used to connect the solutions on the two grids together.

Let us now consider the numerical solution of Burgers' equation in two space dimensions (1.1). We approximate the time derivative in (1.1) using the "implicit

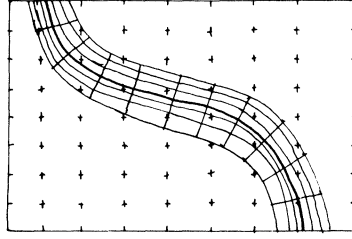


FIG. 5

Euler" approximation, giving

$$\begin{aligned}
 & \varepsilon u_{xx}(x, y, t) + \varepsilon u_{yy}(x, y, t) + f(u(x, y, t))_x + f(u(x, y, t))_y - \frac{1}{k} u(x, y, t) \\
 (2.1) \quad & = -\frac{1}{k} u(x, y, t-k)
 \end{aligned}$$

where $u(\cdot, \cdot, \cdot)$ is an approximation to $U(\cdot, \cdot, \cdot)$, k is the time step and for convenience we have defined $f(u) := -\frac{1}{2}u^2$. For numerical purposes it will also be necessary to restrict the domain of integration to be finite; we choose $-1 \leq x, y \leq 1, 0 \leq t \leq T$ for some finite T . In addition to the initial conditions $u(x, y, 0) = U(x, y, 0)$ we therefore must also specify boundary conditions: $u(x, \pm 1, t)$ given for $-1 \leq x \leq 1$ and $u(\pm 1, y, t)$ given for $-1 \leq y \leq 1$.

A convenient way to implement an implicit difference scheme such as (2.1) is through operator splitting. This reduces the computational problem to a sequence of one-dimensional problems: We introduce an "underlying" coarse mesh $\{x_i, y_j\}_{0,0}^{N,M}$ and solve by difference approximation the equations

$$(2.2a) \quad \varepsilon \tilde{u}_{xx}(x, y_j) + f(\tilde{u}(x, y_j))_x - \frac{1}{k} \tilde{u}(x, y_j) = -\frac{1}{k} u(x, y_j, t-k) \quad \text{for } j=0, 1, \dots, M$$

and

$$(2.2b) \quad \varepsilon u_{yy}(x_i, y, t) + f(u(x_i, y, t))_y - \frac{1}{k} u(x_i, y, t) = -\frac{1}{k} \tilde{u}(x_i, y) \quad \text{for } i=0, 1, \dots, N$$

with initial conditions $u(x, y, 0) = u_0(x, y)$. (This in general introduces an additional error into the solution which is $O(k)$.)

Note that with the obvious notation, each of equations (2.2) is of the form

$$(2.3) \quad \varepsilon \frac{d^2 w}{dz^2} + \frac{d}{dz}[g(w)] + bw = r(z), \quad w = w(z),$$

on $-1 \leq z \leq 1$ with $w(-1), w(1)$ given. This is a singularly perturbed two-point boundary value problem. An extensive analytic theory exists for equations of the form (2.1) and (2.3) (see for example Kevorkian and Cole [7]). Numerical techniques for accurately resolving the features of solutions of problems of the type (2.3) have been developed by B. Kreiss and H. O. Kreiss [9]. We will use a modification of this method due to Brown [1, Chap. 3]. Briefly, each of the nonlinear problems (2.3) is solved by a functional Newton iteration. The linear problems which arise in the iteration are solved using a weighted one-sided difference approximation together with solution-adaptive mesh refinement. The details can be found in Brown [1].

Let us suppose that in the initial conditions $u_0(x, y)$ there is a region of rapid transition oriented obliquely to the mesh. We should begin by solving (2.2a) on each of the lines $y = y_j, j = 1, 2, \dots, M-1$. Because of the rapid transition region, automatic refinement will occur so that the solution of each of those one-dimensional problems would be resolved. We then solve equation (2.2b) on each of the lines $x = x_i, i = 1, 2, \dots, N-1$. Again automatic mesh refinement will occur in the region near the rapid transition. Note, however, that the right-hand side of the equation for $u(x_i, y, t)$ depends on values of the computed solution $\tilde{u}(x_i, y)$ at the previous step. If points are added to the one-dimensional mesh between the coarse mesh lines $y = y_j$, this means that we will need values of \tilde{u} at points where they have not been computed (see Fig. 6). Simple linear interpolation parallel to the coordinate lines will not work well in general because the function being interpolated is not sufficiently smooth. This can be made clear by the following example (refer to Fig. 7): Consider the function $f(x, y)$ defined by

$$f(x, y) = \begin{cases} f_1 & \text{if } x < 2(y - y_1) + x_2, \\ f_2 & \text{otherwise,} \end{cases}$$

where $f_1 \neq f_2$, for $y_1 < y < y_2$. We are interested in determining values $f(\tilde{x}, y)$ for $y_1 < y < y_2$ when we only know the functions $f(x, y_1)$ and $f(x, y_2)$, where $x_1 < \tilde{x} < x_2$. (This corresponds to the situation in Fig. 6 as well.) Linear interpolation along the line $x = \tilde{x}$ will clearly give an inappropriate answer if $|f_1 - f_2|$ is large. Using simple

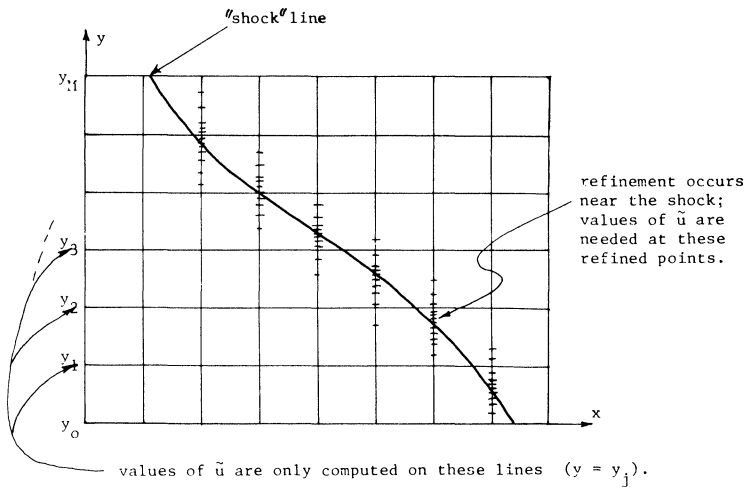


FIG. 6

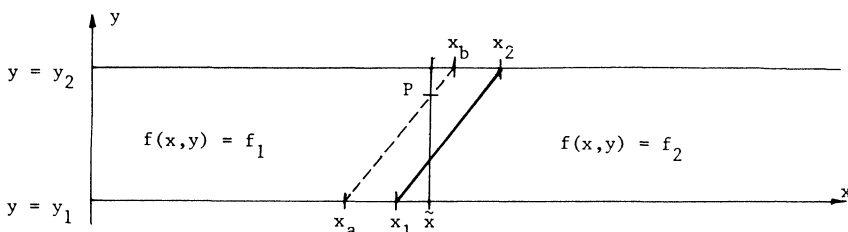


FIG. 7

linear interpolation along $x = \tilde{x}$ we will always get a value between f_1 and f_2 , while the correct value should be *either* f_1 or f_2 . To remedy this problem, we can eliminate the restriction that interpolation always be made along lines of constant x . We can get a reasonable value for f at the point P if, for example, we interpolate linearly along the straight line between (x_a, y_1) and (x_b, y_2) . The interpolation procedure we have implemented is not much different from this simple explanation. However the conditions under which it will give accurate results and the technique for its practical implementation must be considered carefully. This is done in the next section.

3. The interpolation procedure. In this section we consider the problem of how to accurately reconstruct a function defined on only a finite number of appropriately chosen parallel lines, or cross-sections. The function is assumed to consist of large smooth regions separated by one or more nearly one-dimensional regions of singular behavior such as a shock profile. On each cross-section the function has been fully resolved by a sufficiently refined mesh. The method we describe is a completely local procedure. It is second order accurate in terms of the characteristic scale of the smooth parts of the solution (i.e., $O(h^2)$). The shape of any front is approximated by piecewise straight lines and the smooth parts of the solution by piecewise linear functions. Thus in order to interpolate u_0 at the point (x_0, y_0) we only use the values $u_0(x, y_{m+1})$ and $u_0(x, y_m)$, where $y_m \leq y_0 \leq y_{m+1}$. It is clear that the localness of the procedure is a particularly useful feature when the singular domain is topologically complex, for example when there are two or more possibly intersecting shocks.

Consider now the function $u_0(x, y)$ defined on the strip $(-\infty, \infty) \times [0, 1]$. Then we wish to determine $I(u_0)(x_0, y_0)$, the interpolant of the function $u_0(x, y)$ at (x_0, y_0) , from the function values of u_0 for any x and for values of y that belong to the sequence $0 = y_1 < \dots < y_N = 1$.

Our aim is to interpolate using function values obtained from points which can be joined by a smooth curve lying entirely on the same smooth part of the solution. One should note that the points cannot lie at too great a distance from each other. From a practical point of view this means that the shock lines cannot be oriented too nearly parallel to the lines $y = y_j$. This is not a serious restriction, however, because it corresponds to the situation illustrated previously in Fig. 1 and can be taken care of using conventional mesh refinement. In the context of our mesh refinement procedure we would handle this case by simply adding one or more lines to the "underlying coarse mesh" $y = y_j$.

The method depends on the following assumptions about $u_0(x, y)$:

(i) The function should be smooth at distances greater than ε away from a nearly one-dimensional region where there is singular behavior. Here, ε is a positive number much smaller than the natural scale corresponding to changes of $u_0(x, y)$ outside of this region. (Note that this ε is not necessarily the same one as appears in the differential equation (1.1), although if the function being interpolated is a solution of (1.1) then the two are certainly related. In this section we use ε to denote the scale of the region of singular behavior.) If the function u_0 is only known at discrete values, we then assume that away from this singular region a mesh of size h completely resolves the function, where $h \gg \varepsilon$.

(ii) We assume that the singular region is the union of a finite number of smooth curves. In this way the curves can be isolated from each other, and if they intersect, the number of possible intersections is finite.

(iii) We assume that the singular behavior is of the shock type, i.e., a rapid but essentially monotone transition (and not a high frequency oscillation) matching two

has shown that it is best to define this region with two curves. In Fig. 8 these are the lines defined by $[(x_1^0, y_m), (x_0^1, y_{m+1})]$ and $[(x_0^0, y_m), (x_1^1, y_{m+1})]$. For convenience we assume that

$$(3.3) \quad \delta_1 = u_0(x_0, y_{m+1}) - u_0(x_0, y_m) > 0.$$

We now determine x_γ^1 , for $\gamma = 0, 1$, by

$$(3.4a) \quad u_0(x_\gamma^1, y_{m+1}) = u_0(x_0, y_{m+1}) - l_\gamma \delta_1,$$

where we arbitrarily choose $l_0 = \frac{1}{3}$ and $l_1 = \frac{2}{3}$. In a similar way we determine x_γ^0 by

$$(3.4b) \quad u_0(x_\gamma^0, y_m) = y_0(x_0, y_m) + l_\gamma \delta_1.$$

(The slope of the function u_0 in the shock region is of $O(\varepsilon^{-1})$ and so a change in the constants l_γ will only produce an $O(\varepsilon)$ change in the determination of the points x_γ^σ which themselves determine the direction of the shock.)

We now make sure that the points obtained are close enough to each other, that is, we compute

$$(3.5a) \quad (d_t)^2 = (y_{m+1} - y_m)^2 + (x_1^0 - x_0^1)^2,$$

$$(3.5b) \quad (d_b)^2 = (y_{m+1} - y_m)^2 + (x_0^2 - x_1^1)^2.$$

If either d_t or d_b is greater than $2h$, or if it is not possible to determine any of the points x_γ^σ , for $\gamma = 0, 1$ and $\sigma = 0, 1$, then there is not enough information to perform an accurate interpolation. In this case, we need to obtain the values of $u_0(x, \tilde{y})$, where $\tilde{y} = \frac{1}{2}(y_m + y_{m+1})$ and $-\infty < x < \infty$. In the shock problem this is done by going back to the last half time step computation. (See Figs. 9 and 10 for some possible situations in which this procedure asks for more information.)

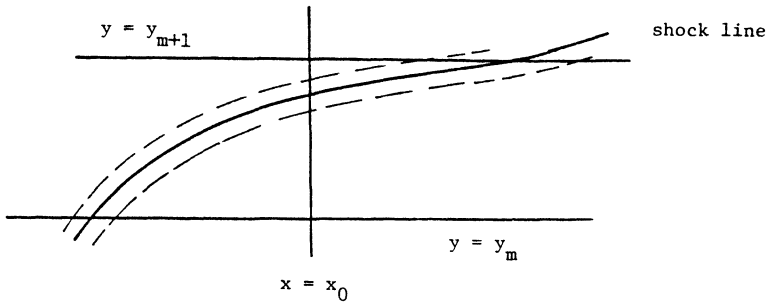


FIG. 9

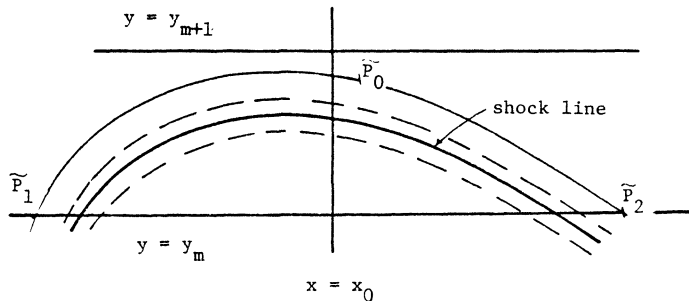


FIG. 10

We must still consider the possibility illustrated in Fig. 11, that is, when two or more singular regions come together at one point. In order for the method to recognize this situation we must also make sure that

$$|u_0(x_1^1 + h, y_{m+1}) - u_0(x_0, y_m)| \leq \beta h,$$

and similarly that

$$|u_0(x_1^0 - h, y_m) - u_0(x_0, y_{m+1})| \leq \beta h.$$

(The example in Fig. 11 would fail the second test because of the presence of shock #2.) If either of these tests fail, we must again ask for more information.

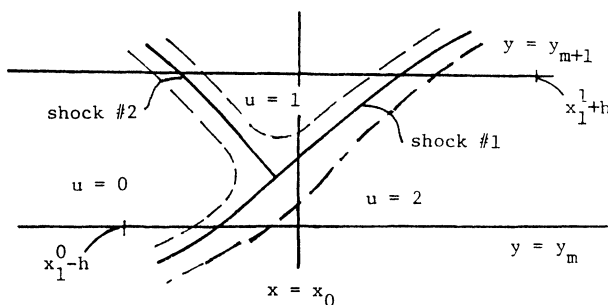


FIG. 11

Having ruled out the anomalous cases we now determine the slope of the segments joining the points that lie on the same part of the smooth solution

$$(3.6a) \quad s_t = \frac{y_{m+1} - y_m}{x_1^0 - x_0^1},$$

$$(3.6b) \quad s_b = \frac{y_{m+1} - y_m}{x_1^1 - x_0^0},$$

and the intersections of the segments with the vertical line through P_m and P_{m+1}

$$(3.7a) \quad y_t = s_t(x - x_0^1) + y_m,$$

$$(3.7b) \quad y_b = s_b(x - x_0^0) + y_m.$$

Defining s in the following way

$$(3.8) \quad s = \begin{cases} s_t & \text{for } y_{m+1} \geq y_0 \geq y_t, \\ \frac{y_0 - y_b}{y_t - y_b} s_t + \frac{y_t - y_0}{y_t - y_b} s_b & \text{for } y_t \geq y_0 \geq y_b, \\ s_b & \text{for } y_b \geq y_0 \geq y_m, \end{cases}$$

we compute the intersection of a straight line through the point (x_0, y_0) with slope s with the top and bottom lines $y = y_m$ and $y = y_{m+1}$

$$(3.9a) \quad \hat{x}_{m+1} = x_0 + \frac{y_{m+1} - y_0}{s},$$

$$(3.9b) \quad \hat{x}_m = x_0 + \frac{y_0 - y_m}{s}.$$

Finally the required interpolated value $I(u_0)(x_0, y_0)$ is determined by linear interpolation using the function values of u_0 at the points (\hat{x}_m, y_m) and (\hat{x}_{m+1}, y_{m+1}) :

$$(3.10) \quad I(u_0)(x_0, y_0) = \frac{y_0 - y_m}{y_{m+1} - y_m} u_0(\hat{x}_{m+1}, y_{m+1}) + \frac{y_{m+1} - y_0}{y_{m+1} - y_m} u_0(\hat{x}_m, y_m).$$

Case II.b. If $\delta u_0(x_0, m) \leq \beta h$, we then have a shock close to either point (or close to both points). Introduce

$$\tilde{\delta}(\tilde{x}, m, n) = \max(|u_0(\tilde{x}, y_m) - u_0(\tilde{x} + h, y_n)|, |u_0(\tilde{x}, y_m) - u_0(\tilde{x} - h, y_n)|);$$

and define $\delta_2 = \max(\tilde{\delta}(x_0, m, m+1), \tilde{\delta}(x_0, m+1, m))$. If $\delta_2 \leq \beta h$ then we need more information (see the discussion following equations (3.5)). Otherwise we proceed as in Case II.a looking for either a vertical or an oblique or curved shock, as in Fig. 4. If it is not possible to find any such structure we again need extra information.

Notice that the interpolation procedure does not produce a continuous function of (x_0, y_0) ; this is because a different interpolation method is used in regions near the shock than in regions away from the shock. Recall that near the shock, linear interpolation along lines essentially parallel to the shock is used while away from the shock, linear interpolation parallel to the underlying coarse grid lines is used. The interpolation method chosen changes discontinuously as a function of the distance from the region of singularity. This is of no practical importance, however, because the interpolant is needed and computed at only a discrete set of x values.

In the cases corresponding to Figs. 9 and 10 the interpolation method will fail and will ask for extra lines $y = \text{const.}$ to be added until δy , the distance between two consecutive horizontal lines, is $O(\varepsilon)$, the width of the shock. Note that this failure is not due to the fact that we are representing the front with straight line segments. If we were to use a higher order fitting method to represent the front and even if we assume that its shape is known exactly, we would still need to add these extra horizontal lines. This is explained as follows: In order to find the value of the interpolant at $P_0 = (x_0, y_0)$ (Fig. 10), we perform some interpolation along a curve parallel to the front. The interpolation formula will link values of the functions at points such as \tilde{P}_1 and \tilde{P}_2 , and possibly points on other horizontal lines. The distance between these points is $O(\delta y^{1/2})$ in the cases of Figs. 9 and 10. This implies that if we only restrict ourselves to interpolate using points that are $O(h)$ apart from each other then we must restrict δy to be $O(h^2)$. Hence, a higher order fitting method will reduce the number of operations when compared to the second order fitting method we have described, but there will be no saving in the number of coarse mesh lines needed.

4. The interpolation error. In order to understand the interpolation error, we consider the problem corresponding to $\varepsilon = 0$ (that is, an actual discontinuity) and to only one shock. In this case $x_0^0 = x_1^0 = x_b$, $x_0^1 = x_1^1 = x_t$ and $s_t = s_b = s$ where x_b corresponds to the intersection of the shock line with the line $y = y_m$ and similarly for x_t (see Fig. 8). Without any loss of generality we can consider $x_b < x_t$. In this case the interpolant evaluated at (x_0, y_0) (for $y_m \leq y_0 \leq y_{m+1}$) is defined by equation (3.2) when $x_0 \geq x_t$ or $x_0 \leq x_b$, and by equation (3.10) when $x_b < x_0 < x_t$.

It is not possible to obtain an error formula in the maximum norm. The existence of such an error formula would imply that it is possible to determine the shape of an arbitrary curve on the plane from a finite number of its points. Nevertheless we have the following obvious local error estimates (see Fig. 12): the error is $O(\delta y^2)$ away from the shock (region I), $O(\delta y^2 \sqrt{1+s^{-2}})$ near the shock region (region II), and $O(1)$ between the shock and the chord $[(x_b, y_m), (x_t, y_{m+1})]$ (region III). The area of this last

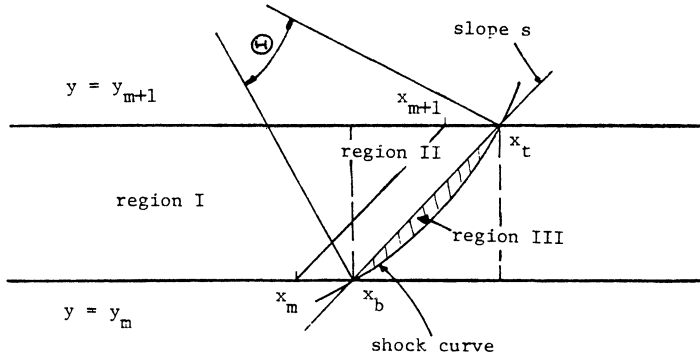


FIG. 12

region is $A = \frac{1}{2}\kappa^{-2}(\theta - \sin(\theta))(1 + O(\delta y))$ where κ is a value characteristic of the curvature of the shock front when the front lies between the lines $y = y_m$ and $y = y_{m+1}$, and θ is the change in the angle of the tangents of the shock front as the shock moves from $y = y_m$ to $y = y_{m+1}$. The angle θ is determined by $\sin(\frac{1}{2}\theta) = \frac{1}{2}\kappa \delta y \sqrt{1 + s^{-2}}$.

In order to have an accurate interpolation, the shape of the front has to be resolved. We can assume that this has been achieved when A_{III} , the total area of regions of type III, is $O(h^2)$. Now, when θ is small and $|s| > \delta y$ we have that $A \approx \kappa(\sqrt{1 + s^{-2}} \delta y)^3$. Thus in order to resolve the shock we need $A \delta y / s = O(h^2)$. In this way when $s = O(1)$ and when the shock is a smooth curve (i.e., when we have an upper bound for κ), the condition on the area amounts to $\delta y / s$ being $O(h)$. This relation between the orientation of the shock and the distance between two consecutive horizontal lines was enforced by making d_t and d_b smaller than $2h$. On the other hand when $|s| < \delta y$, we have that $A \approx \delta y$; hence, in this case we need $\delta y = O(h^2)$. This implies that we should stop adding extra horizontal lines when δy is $O(h^2)$.

5. Numerical examples. In this section we present some numerical examples designed to test the interpolation procedure discussed in the last two sections. In particular we include some examples where the interpolation procedure has been used in conjunction with operator splitting to solve (1.1) for two sets of initial data.

Example 1. To test the performance of the interpolation procedure, we first used it to interpolate two known functions containing a region of rapid transition and compared the results of the interpolation with the original function. The functions considered were

$$(5.1) \quad u_0(x, y) = \tanh((y - \frac{1}{2}x^2 - \frac{1}{2}x)/\varepsilon)$$

and

$$(5.2) \quad u_0(x, y) = \tanh((y - x)/\varepsilon) - \frac{1}{4} \sin(\pi(x + y))$$

on $0 \leq x \leq 1$ with $\varepsilon = .02$. The function (5.1) models the case where the shock line is curved; (5.2) models the case where the smooth part of the function is not constant. We began by calculating the function u_0 at specified points $x = x_k^{(j)} \in [0, 1]$, $k = 1, 2, \dots, n_j$ along uniformly spaced parallel lines $y = y_j$, $j = 0, 1, \dots, N$, $N = 20$. We also calculated the function along the additional boundary lines of $0 \leq x, y \leq 1$ i.e., $x = 0$, $x = 1$. The interpolation points were specified in such a way that the function was "resolved" on each cross-section.

In the first step of the interpolation, the function $u_0(x, y)$ was interpolated onto $N - 1$ uniformly spaced cross-sections $x = x_l$, $l = 1, 2, \dots, N - 1$ (perpendicular to the

original cross-sections). Again this was done at points along these cross-sections that were chosen so that the function would be well-resolved. We denoted the resulting function by $u_1(x, y)$.

This process was then repeated, interpolating the function $u_1(x, y)$ back onto the original cross-sections $y = y_j$, obtaining a function $u_2(x, y)$ defined at points $x = x_k^{(j)}$ as above.

Let $e(x, y) = u_2(x, y) - u_0(x, y)$; then we introduce the maximum norm, the L_1 -norm and L_2 -norm of the error by

$$\begin{aligned}
 \|e(x, y)\|_\infty &= \max_{x_k^{(j)}, y_j} |e(x_k^{(j)}, y_j)|, \\
 \|e(x, y)\|_2^2 &= \sum_{j,k} (e(x_k^{(j)}, y_j))^2 h_{j,k} / N, \\
 \|e(x, y)\|_1 &= \sum_{j,k} |e(x_k^{(j)}, y_j)| h_{j,k} / N,
 \end{aligned}
 \tag{5.3}$$

where

$$h_{j,k} = \begin{cases} x_1^{(j)} - x_0^{(j)} & \text{if } k = 0, \\ x_k^{(j)} - x_{k-1}^{(j)} & \text{if } k \neq 0, \end{cases}
 \tag{5.4}$$

is the “meshwidth” associated with the k th interval on the j th line $y = y_j$. The errors observed for the functions (5.1) and (5.2) were

$$\|e\|_\infty = .056, \quad \|e\|_2 = .0084, \quad \|e\|_1 = .0032,
 \tag{5.5}$$

and

$$\|e\|_\infty = .024, \quad \|e\|_2 = .0047, \quad \|e\|_1 = .0022,
 \tag{5.6}$$

respectively.

Figures 13 and 14 show the results of this interpolation graphically for the functions (5.1) and (5.2), respectively. Figures a show the original function, Figs. b the interpolant and Figs. c the interpolation error for each of these examples. The scale has been magnified somewhat in the error plots. We see that the function and its interpolant

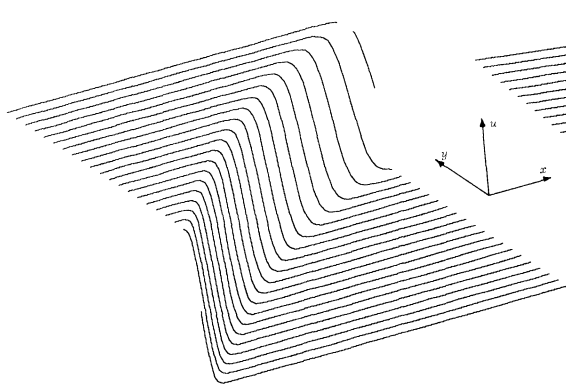


FIG 13a. *Function to be interpolated.*

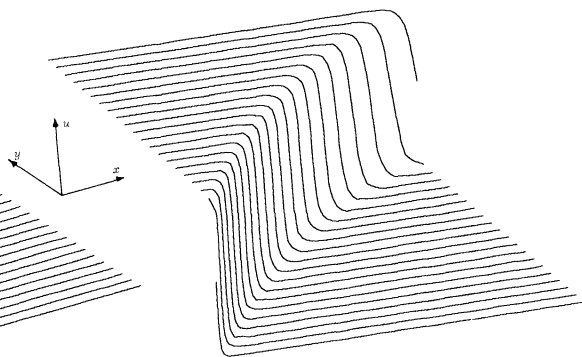
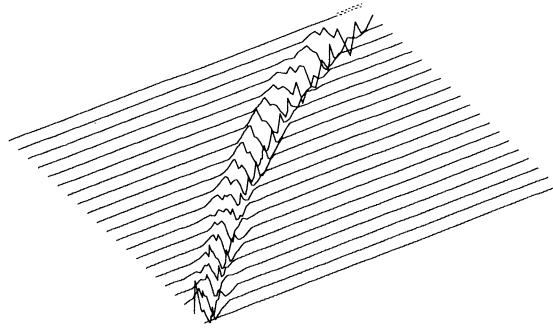
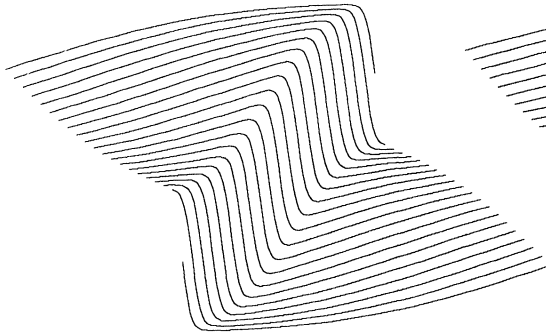
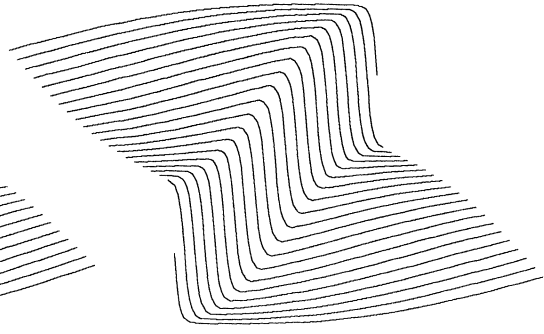
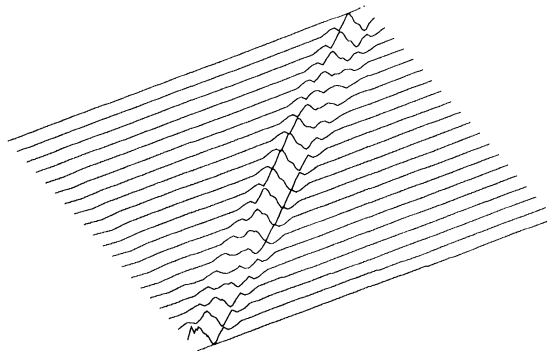


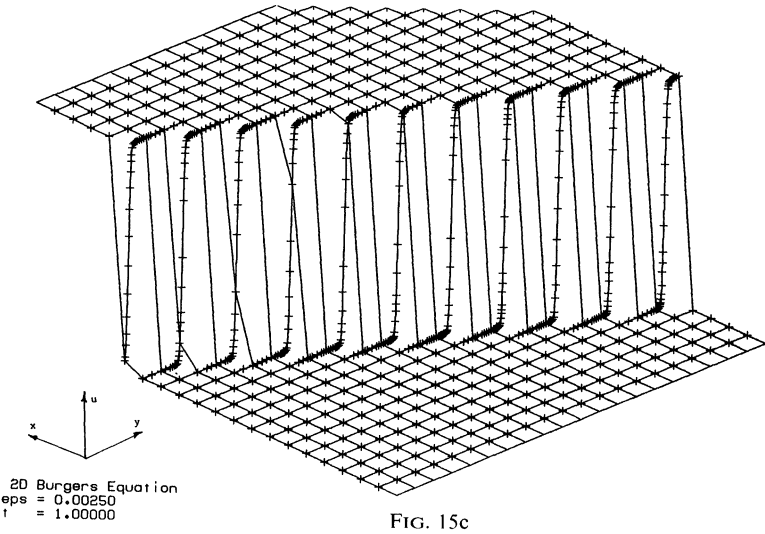
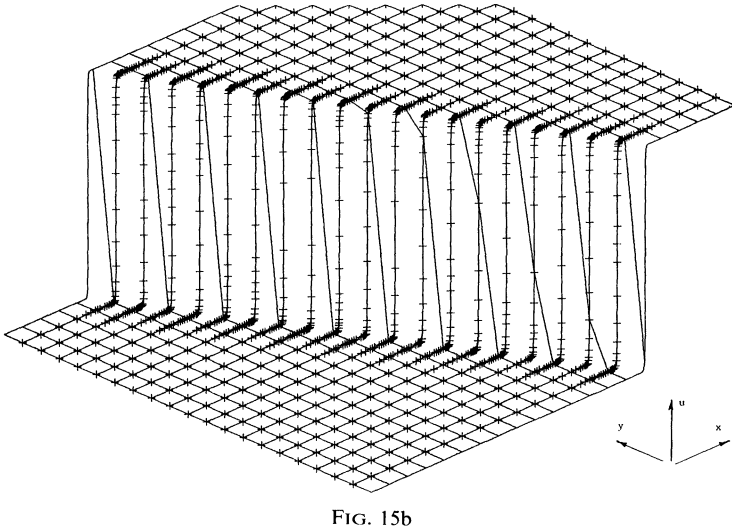
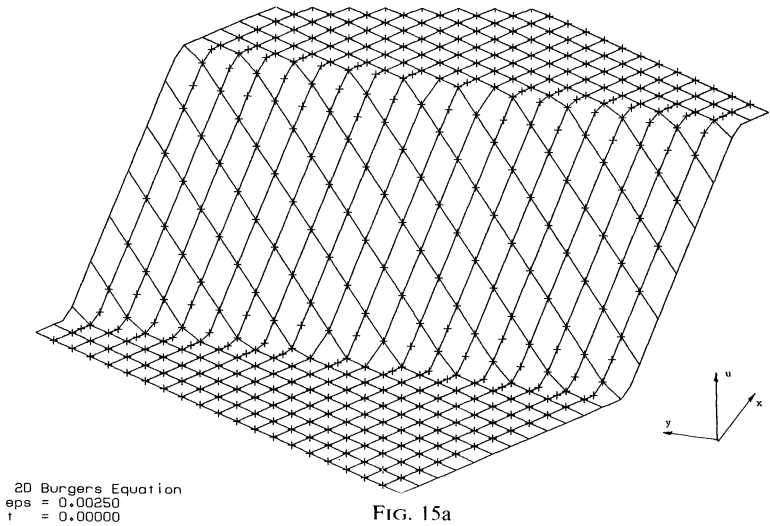
FIG. 13b. *The interpolated function.*

FIG. 13c. *The interpolation error.*

are difficult to distinguish; in particular, the transition region in the functions is quite well resolved even after being interpolated twice. As we would expect, the error is confined to the region of singularity in the function.

Example 2. We used our interpolation procedure in conjunction with operator splitting to solve the two-dimensional Burgers' equation with initial conditions that result in stationary rapid transitions oriented obliquely to the mesh.

FIG. 14a. *The function to be interpolated.*FIG. 14b. *The interpolated function.*FIG. 14c. *The interpolation error.*



TWO-DIMENSIONAL MESH REFINEMENT

2D Burgers Equation $\epsilon = 0.002500$ $t = 0.000000$

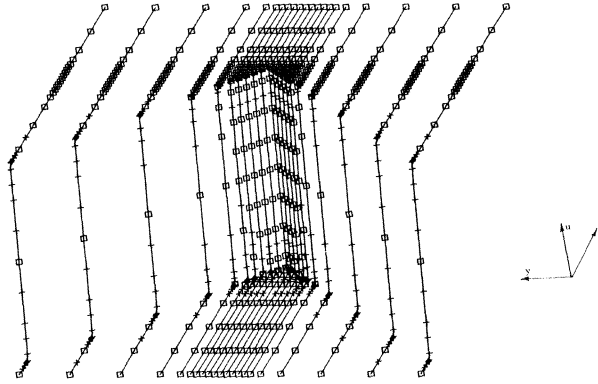


FIG. 16a

2D Burgers Equation $\epsilon = 0.002500$ $t = 0.200000$

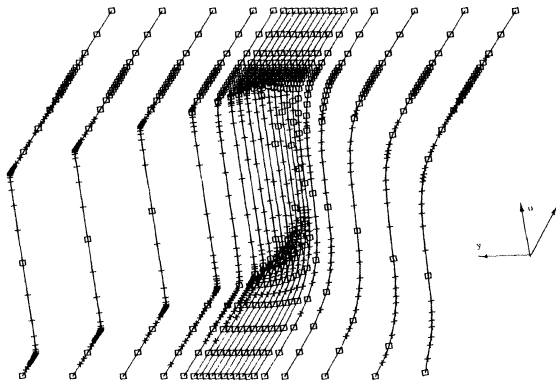


FIG. 16b

2D Burgers Equation $\epsilon = 0.002500$ $t = 0.200000$

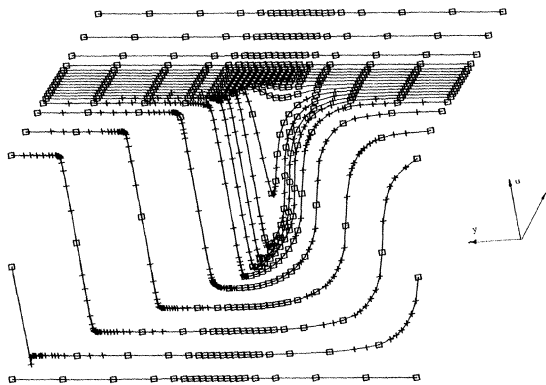


FIG. 16c

2D Burgers Equation $\epsilon = 0.002500$ $t = 1.000000$

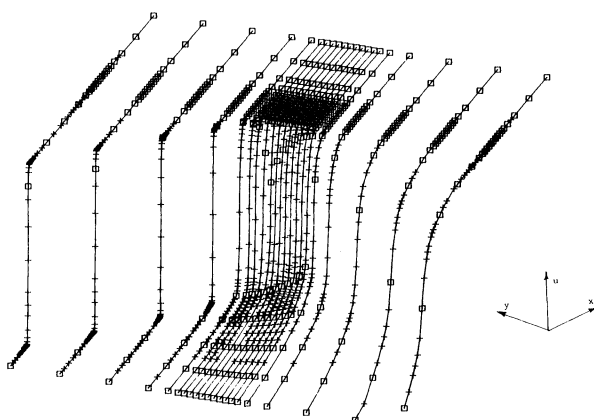


FIG. 16d

2D Burgers Equation $\epsilon = 0.002500$ $t = 1.000000$

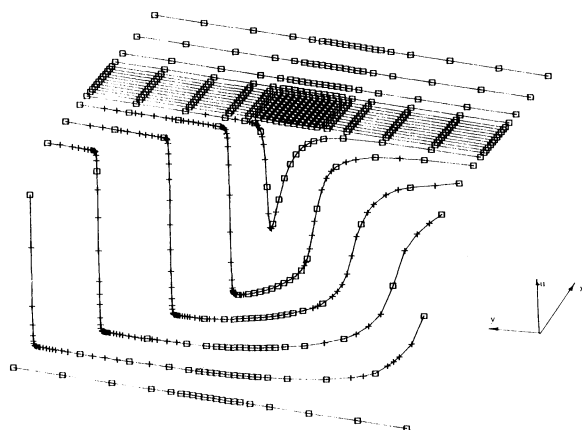


FIG. 16e

Figures 15 show the initial data and solution at time $t = 1$ for a computation using this method. The initial data (Fig. 15a) is a ramp oriented obliquely with respect to the mesh connecting the constant values $u = \pm 1$. Fig. 15b shows the solution after the last sweep in x and Fig. 15c shows the solution after the last sweep in y . The plus signs “+” indicate the locations of the mesh points in the final refined mesh. Lines are also drawn in the direction perpendicular to the sweep direction (e.g., in the y -direction in Fig. 15a) to indicate the location of the underlying coarse mesh. (Note that Fig. 15c is reversed in orientation from the other two plots in this series.)

Figures 16 show the initial data and computed solutions at time $t = 0.2$ and time $t = 1$ for another example using this method. The coarse mesh in this case was not a uniform one, but was finer near the center of the domain where the corner of the “wedge” occurs. This was done in an attempt to resolve that corner. The initial data also consist of ramps connecting the two constant states $u = \pm 1$. The two ramps are oriented in such a way that the one on the left evolves into a shock while the one on the right forms a contact discontinuity. Because of the dissipative terms in Burgers’

equation, of course, the shock has finite width, and the contact discontinuity becomes wider with time. In this series of plots, the orientation is the same for all sweeps shown. The meshpoints are indicated with small squares and plus signs. The squares denote meshpoints that lie on the underlying coarse mesh. Figures 16b and 16d are the solutions after the x -sweep at $t = 0.2$ and $t = 1.0$, respectively; Figs. 16c and 16e show the solutions after the corresponding y -sweeps. In all the computations presented in this section, $\varepsilon = 1/400$, and the time step was $k = 1/20$. Note that, in particular, the intended objective of this method, to resolve steady two dimensional rapid transitions, has been realized.

Acknowledgment. We would like to thank Michael Naughton for carefully reading the manuscript and making several helpful comments.

REFERENCES

- [1] D. L. BROWN, *Solution adaptive mesh procedures for the numerical solution of singular perturbation problems*, Ph.D. thesis, Dept. Applied Mathematics, California Institute of Technology, Pasadena.
- [2] M. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equation*, J. Comp. Phys. to appear.
- [3] A. J. CHORIN, *Random choice solution of hyperbolic systems*, J. Comp. Phys., 22 (1976), pp. 517-533.
- [4] B. ENGQUIST AND S. OSHER, *One-sided difference approximations for nonlinear conservation laws*, Math. Comp., 36 (1981) pp. 321-352.
- [5] W. D. GROPP, *A test of moving mesh refinement for 2-D scalar hyperbolic equations*, this Journal, 1 (1980), pp. 191-197.
- [6] A. HARTEN AND P. D. LAX, *A random choice finite difference scheme for hyperbolic conservation laws*, SIAM J. Numer. Anal., 18 (1981), pp. 289-315.
- [7] J. KEVORKIAN AND J. COLE, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, New York, 1981.
- [8] B. KREISS, *Construction of curvilinear grids*, this Journal, 4 (1983), pp. 270-279.
- [9] B. KREISS AND H. O. KREISS, *Numerical methods for singular perturbation problems*, SIAM J. Numer. Anal., 18 (1981), pp. 262-276.
- [10] H. O. KREISS, personal communication, 1980.
- [11] R. J. MILLER, S. K. DOSS AND K. MILLER, *The moving finite element method: applications to general partial differential equations with multiple large gradients*, J. Comp. Phys., 40 (1981), pp. 202-249.
- [12] S. OSHER AND F. SOLOMON, *Upwind difference schemes for hyperbolic systems of conservation laws*, Math. Comp., 38 (1982), pp. 339-374.